**#342** [Soarece](https://www.pbinfo.ro/?pagina=probleme&id=342) Se dă o tablă dreptunghiulară formată din *n* linii și *m* coloane, definind *n\*m*zone, unele dintre ele fiind libere, altele conținând obstacole. În zona aflată la poziția *is*, *js* se află un șoarece care se poate deplasa pe tablă trecând din zona curentă în zona învecinată cu aceasta pe linie sau pe coloană. Scopul sau este să ajungă la o bucată de brânză aflată în zona de la poziția *ib*, *jb*, fără a părăsi tabla, fără a trece prin zone care conțin obstacole și fără a trece de două ori prin aceeași zonă. Determinați câte modalități prin care șoarecele poate ajunge de la poziția inițială la cea a bucății de brânză există.

# Date de intrare Fişierul de intrare *soarece.in* conţine pe prima linie numerele *n m*, separate printr-un spațiu. Următoarele *n* linii conțin câte *m* valori *0* sau *1*, separate prin exact un spațiu, care descriu tabla – valoarea *0* reprezintă o zonă liberă, valoarea *1*reprezintă o zonă ocupată cu un obstacol. Pe linia *n+2* se află *4* numere separate prin exact un spațiu, reprezentând *is js ib jb*.

# Date de ieşire Fişierul de ieşire *soarece.out* va conţine pe prima linie numărul *S*, reprezentând numărul de modalități prin care șoarecele poate ajunge de la poziția inițială la cea a bucății de brânză.

# Restricţii şi precizări *1 ≤ n,m ≤ 10, 1 ≤ is,ib ≤ n*, *1 ≤ js,jb ≤ m,* poziția șoarecelui și cea a bucății de brânză nu sunt identice și sunt libere

# Exemplu *soarece.in soarece.out*

6 7 8

0 0 0 0 0 0 0

0 1 1 1 1 0 0

0 0 0 0 1 1 0

0 1 1 0 1 0 0

0 1 1 0 1 0 1

0 0 0 0 0 0 0

4 1 2 6

#include <fstream> using namespace std; ifstream fin("soarece.in"); ofstream fout("soarece.out");

int nr=0,a[12][12],viz[12][12],n,m,is,js,ib,jb; int dx[4]= {-1,0,1,0}; int dy[4]= {0, 1,0,-1};

void bordare() { int i,j;

for(i=0;i<=n+1;i++) a[i][0]=a[i][m+1]=1;

for(j=0;j<=m+1;j++) a[0][j]=a[n+1][j]=1; }

void **soarece**(int x,int y) { int k,xx,yy;

**if(x==ib && y==jb) nr++;**

**else for(k=0; k<4; k++)**

**{ xx=x+dx[k]; yy=y+dy[k];**

**if(a[xx][yy]==0 && viz[xx][yy]==0) { viz[xx][yy]=1; soarece(xx,yy); viz[xx][yy]=0; }**

**} }**

int main() { int i,j; fin>>n>>m;

for(i=1; i<=n; i++) for(j=1; j<=m; j++) fin>>a[i][j];

fin>>is>>js>>ib>>jb;

bordare();

viz[is][js]=1;

soarece(is,js);

fout<<nr; }

#include <fstream> using namespace std; ifstream fin("soarece.in"); ofstream fout("soarece.out");

void citire(); void bordare(); void afisare(); void bkt(int x, int y);

int dx[]={0,0,1,0,-1}; int dy[]={0,-1,0,1,0}; int a[12][12], viz[12][12], sx, sy, bx, by, n, m, drum;

int main() { citire(); bordare(); viz[sx][sy]=1;

bkt(sx, sy); afisare(); return 0; }

void **bkt**(int x, int y) { if(x==bx&&y==by) drum++;

else { int i;

for(i=1;i<=4;i++)

if(a[x+dx[i]][y+dy[i]] == 0 && viz[x+dx[i]][y+dy[i]] == 0)

{ viz[x+dx[i]][y+dy[i]] = 1; bkt(x+dx[i], y+dy[i]); viz[x+dx[i]][y+dy[i]] = 0; } } }

void citire() { int i, j; fin >> n >> m;

for(i=1;i<=n;i++)

for(j=1;j<=m;j++) fin >> a[i][j];

fin >> sx >> sy >> bx >> by; }

void **bordare**() { int i, j;

for(i=0;i<=n+1;i++) { a[i][0]=a[i][m+1]=1; a[0][i]=a[n+1][i]=1; } }

void afisare() { fout << drum; int cioc = 0; if(cioc == 1)

{ int i, j;

for(i=0;i<=n+1;i++)

{ for(j=0;j<=m+1;j++) fout << a[i][j] << ' ';

fout << endl; }

}

}